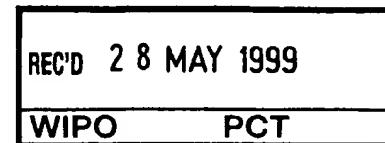


日本国特許庁 31.03.99
 PATENT OFFICE
 JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
 いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
 with this Office.

出願年月日
 Date of Application: 1998年 3月31日



出願番号
 Application Number: 平成10年特許願第103720号

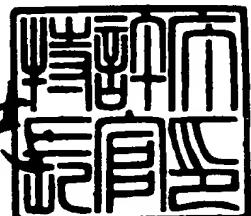
出願人
 Applicant(s): セイコーエプソン株式会社

PRIORITY
 DOCUMENT
 SUBMITTED OR TRANSMITTED IN
 COMPLIANCE WITH RULE 17.1(a) OR (b)

1999年 5月14日

特許庁長官
 Commissioner,
 Patent Office

佐藤 建



出証番号 出証特平11-3028509

【書類名】 特許願
【整理番号】 EP115501
【提出日】 平成10年 3月31日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 11/22
G06F 11/28
【発明の名称】 マイクロコンピュータ、電子機器及びデバッグシステム
【請求項の数】 6
【発明者】
【住所又は居所】 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内
【氏名】 土方 陽一
【特許出願人】
【識別番号】 000002369
【氏名又は名称】 セイコーエプソン株式会社
【代理人】
【識別番号】 100090479
【弁理士】
【氏名又は名称】 井上 一
【電話番号】 03-5397-0891
【代理人】
【識別番号】 100090387
【弁理士】
【氏名又は名称】 布施 行夫
【電話番号】 03-5397-0891
【代理人】
【識別番号】 100090398
【弁理士】
【氏名又は名称】 大渕 美千栄

【電話番号】 03-5397-0891

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9402500

【書類名】 明細書

【発明の名称】 マイクロコンピュータ、電子機器及びデバッグシステム

【特許請求の範囲】

【請求項1】 ユーザーモード及びデバッグモードを有するマイクロコンピュータであって、

ユーザー モード及びデバッグモードが切り替え可能に形成され、それぞれのモードにおいて命令の実行処理を行う中央処理ユニットと、

ユーザー モードでは使用されない端子を介して強制ブレークの入力がされると、前記中央処理ユニットをユーザー モードからデバッグモードに切り替える切り替え手段とを含むことを特徴とするマイクロコンピュータ。

【請求項2】 請求項1において、

前記マイクロコンピュータが、

オンチップデバッグ機能を有し、オンチップデバッグを行うためのデバッグ情報を外部のデバッグツールと送受信するための通信ラインが接続されるデバッグ用通信端子を含み、

前記デバッグ用端子を介して強制ブレークの入力がされることを特徴とするマイクロコンピュータ。

【請求項3】 請求項2において、

前記マイクロコンピュータは、

マイクロコンピュータの外部に設けられデバッグコマンドを少なくとも1つのプリミティブコマンドに変換するための処理を行う第2のモニタ手段との間でデータを送受信し、実行するプリミティブコマンドを前記第2のモニタ手段からの受信データに基づいて決定し、決定したプリミティブコマンドを実行するための処理を行う第1のモニタ手段とを含み、

前記データを半2重の双方向通信で送受信を行うための1本の通信ラインが接続されるデバッグ用端子を含み、

前記中央処理ユニットは、

ユーザー モードにおいてはユーザープログラムの実行処理を行い、

デバッグモードにおいて前記プリミティブコマンドの実行処理を行い、

前記切り替え手段は、

前記デバッグ用端子を介して強制ブレークの入力がされると、前記中央処理ユニットをユーザーモードからデバッグモードに切り替えることを特徴とするマイクロコンピュータ。

【請求項4】 請求項1乃至3のいずれかにおいて、

外部のデバッグツールが接続されていない状態には強制ブレークの入力が行われる端子をハイ又はローのいずれかである第一のレベルに保持する手段を含み、

前記中央処理ユニットは、

リセット時に前記強制ブレークの入力が行われる端子が前記第一のレベルである場合にはユーザーモードで実行を開始し、リセット時に前記強制ブレークの入力が行われる端子が前記第一のレベルでない場合にはデバッグモードで実行を開始することを特徴とするマイクロコンピュータ。

【請求項5】 請求項1乃至4のいずれかのマイクロコンピュータと、

前記マイクロコンピュータの処理対象となるデータの入力源と、

前記マイクロコンピュータにより処理されたデータを出力するための出力装置とを含むことを特徴とする電子機器。

【請求項6】 マイクロコンピュータを含むターゲットシステムのためのデバッグシステムであって、

ホストシステムが発行したデバッグコマンドを少なくとも1つのプリミティブコマンドに変換するための処理を行う第2のモニタ手段と、

前記第2のモニタ手段との間でデータを送受信し、実行するプリミティブコマンドを前記第2のモニタ手段からの受信データに基づいて決定し、決定したプリミティブコマンドを実行するための処理を行う第1のモニタ手段と、

ユーザーモード及びデバッグモードが切り替え可能に形成され、ユーザーモードにおいて前記プリミティブコマンドの実行処理を行う中央処理ユニットと、

前記中央処理ユニットを含むチップに設けられ、前記データを半2重の双方向通信を行うための1本の通信ラインが接続されるデバッグ用端子と、

前記デバッグ用端子を介して強制ブレークの入力がされると、前記中央処理ユ

ニットをユーザーモードからデバッグモードに切り替える切り替え手段とを含むことを特徴とするデバッグシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、マイクロコンピュータ、マイクロコンピュータを含む電子機器、及びデバッグシステムに関する。

【0002】

【背景技術及び発明が解決しようとする課題】

近年、ゲーム装置、カーナビゲーションシステム、プリンタ、携帯情報端末などの電子機器に組み込まれ、高度な情報処理を実現できるマイクロコンピュータに対する需要が高まっている。このような組み込み型のマイクロコンピュータは、通常、ターゲットシステムと呼ばれるユーザボードに実装される。そして、このターゲットシステムを動作させるソフトウェアの開発を支援するためにICE (In-Circuit Emulator) と呼ばれるソフトウェア開発支援ツールが広く使用されている。

【0003】

さて、このようなICEとしては、従来、図1 (A) に示すようなCPU置き換え型と呼ばれるICEが主流を占めていた。このCPU置き換え型ICEでは、デバッグ時にターゲットシステム300からマイクロコンピュータ302を取り外し、その代わりにデバッグツール304のプローブ306を接続する。そして、このデバッグツール304に、取り外したマイクロコンピュータ302の動作をエミュレートさせる。また、このデバッグツール304に、デバッグのために必要な種々の処理を行わせる。

【0004】

しかしながら、このCPU置き換え型ICEには、プローブ306のピン数が多くなると共にプローブ306の線308が増えるという欠点があった。このため、マイクロコンピュータ302の高周波数動作をエミュレートすることが困難になる（例えば33MHz程度が限界）。またターゲットシステム300の設計

も困難になる。更に、マイクロコンピュータ302を実装して動作させる実動作時とデバッグツール304でマイクロコンピュータ302の動作をエミュレートするデバッグモード時とで、ターゲットシステム300の動作環境（信号のタイミング、負荷条件）が変化してしまう。またこのCPU置き換え型ICEには、マイクロコンピュータが異なれば、たとえそれが派生品であっても、設計が異なるデバッグツールや、ピン数やピンの位置が異なるプローブを使用しなければならないという問題もあった。

【0005】

一方、このようなCPU置き換え型ICEの欠点を解消するものとして、ICEと同じ機能を実現するためのデバッグ用のピンと機能を量産チップ上に実装するタイプのICEが知られている。このようなデバッグ機能実装型ICEにおいては、通常ユーザーモードとデバッグモードを有している。そしてユーザーモードにおいて、ユーザープログラムが実行され、デバッグモードにおいてはデバッグ用プログラムが実行される。

【0006】

ユーザーモード及びデバッグモードを有するマイクロコンピュータにおいては、デバッグ時にユーザーモードでユーザープログラムの実行中にプログラムの暴走や無限ループにおちいることがある。このような場合、ユーザーモードからデバッグモードに強制的に移行させるための手段が必要となる。このため、強制ブレークの機能が設けられており、通常強制ブレークを実現するための専用の外部端子を設けている。

【0007】

このように強制ブレークを実現するための外部入力端子を設けるとパッケージのピン数が増える。しかし、デバッグ時にのみ必要でエンドユーザーにとっては不要な端子はできるかぎり少ないほうが好ましい。

【0008】

一方このようなデバッグ機能実装型ICEにおいては 外部（チップ以外のハード等）との通信が不可欠である。しかし外部のデバッグツールと送受信はデバッグモード時のみに行われる。このためユーザーモード時にはこの送受信用端子

は使用されない。

【0009】

強制ブレークを入力するための端子やデバッグモード時にのみ使用する端子は
エンドユーザーにとっては不要な物であるためより少ないほうが好ましい。

【0010】

本発明は、以上のような技術的課題に鑑みてなされたものであり、その目的とするところは、デバッグ用のピンと機能を量産チップ上に実装するタイプのIC
Eにおいて、強制ブレークを入力するための端子及びデバッグモード時にのみ使用する端子等のエンドユーザーにとっては不要な端子をより節約したマイクロコンピュータ、これを含む電子機器、及びデバッグシステムを提供することにある。

【0011】

【課題を解決するための手段】

上記課題を解決するために本発明は、ユーザー mode 及びデバッグモードを有するマイクロコンピュータであって、ユーザー mode 及びデバッグモードが切り替え可能に形成され、それぞれのモードにおいて命令の実行処理を行う中央処理ユニットと、ユーザー mode では使用されない端子を介して強制ブレークの入力がされると、前記中央処理ユニットをユーザー mode からデバッグモードに切り替える切り替え手段とを含むことを特徴とする。

【0012】

ここにおいて強制ブレークとは、ユーザー mode からデバッグモードに強制的に移行させることをいう。

【0013】

本発明によればユーザー mode では使用されない端子、例えばデバッグモードでのみ使用される端子を用いて強制ブレークの入力を行う。従って強制ブレーク専用端子を設ける必要が無いため、マイクロコンピュータの端子の節約となり、ユーザーが使用可能な端子をより多く確保することができる。

【0014】

また本発明は、前記マイクロコンピュータが、オンチップデバッグ機能を有し

、オンチップデバッグを行うためのデバッグ情報を外部のデバッグツールと送受信するための通信ラインが接続されるデバッグ用通信端子を含み、前記デバッグ用端子を介して強制ブレークの入力がされることを特徴とする。

【0015】

オンチップデバッグ機能を有するマイクロコンピュータでは通常外部（チップ以外のハード等）との通信が必要となる。しかし外部のデバッグツールと送受信はデバッグモード時のみに行われ、ユーザー モード時にはこのデバッグ用端子は使用されない。

【0016】

一方強制ブレークの入力はユーザー モードからデバッグモードへ切り替えるための入力であるためユーザー モードでのみ発生する。

【0017】

このようにデバッグ情報と強制ブレーク入力は異なるモードでしか発生しないため、同一の端子を共用してもそれを判別可能で混同は生じない。

【0018】

本発明によれば、オンチップデバッグに必要な端子と強制ブレークを入力するための端子を共用するため、ユーザーが使用可能な端子をより多く確保することができる。

【0019】

また本発明は、前記マイクロコンピュータが、マイクロコンピュータの外部に設けられデバッグコマンドを少なくとも1つのプリミティブコマンドに変換するための処理を行う第2のモニタ手段との間でデータを送受信し、実行するプリミティブコマンドを前記第2のモニタ手段からの受信データに基づいて決定し、決定したプリミティブコマンドを実行するための処理を行う第1のモニタ手段とを含み、前記データを半2重の双方向通信で送受信を行うための1本の通信ラインが接続されるデバッグ用端子を含み、前記中央処理ユニットは、ユーザー モードにおいてはユーザープログラムの実行処理を行い、デバッグモードにおいて前記プリミティブコマンドの実行処理を行い、前記切り替え手段は、前記デバッグ用端子を介して強制ブレークの入力がされると、前記中央処理ユニットをユーザー

モードからデバッグモードに切り替えることを特徴とする。

【0020】

本発明によれば、マイクロコンピュータの外部に設けられた第2のモニタ手段が、ホストシステム等が発行したデバッグコマンドをプリミティブコマンドに変換（分解）するための処理を行う。そして、第1のモニタ手段は、この第2のモニタ手段からデータを受信し、この受信データに基づいて決定されたプリミティブコマンドを実行するための処理を行う。本発明によれば、第1のモニタ手段の処理を実行するためのモニタプログラムに、各デバッグコマンドを実行するための複雑なルーチンを持たせる必要がなくなる。したがって、モニタプログラムの命令コードサイズを格段に小さくできるようになり、小さなハードウェア規模でオンチップデバッグ機能を実現できるようになる。

【0021】

またマイクロコンピュータのデバッグ用端子（ピン）数を減らすことが可能となり、マイクロコンピュータの低コスト化を図れるようになる。

【0022】

また本発明は、外部のデバッグツールが接続されていない状態には強制ブレークの入力が行われる端子をハイ又はローのいずれかである第一のレベルに保持する手段を含み、前記中央処理ユニットは、リセット時に前記強制ブレークの入力が行われる端子が前記第一のレベルである場合にはユーザー mode で実行を開始し、リセット時に前記強制ブレークの入力が行われる端子が前記第一のレベルでない場合にはデバッグモードで実行を開始することを特徴とする。

【0023】

本発明によれば、外部のデバッグツールが接続されていない場合にはプルアップ等によりハイ又はローのいずれかである示す第一のレベルに保持されている。従って前記デバッグ用端子を接続することにより、前記強制ブレークの入力が行われる端子が前記第一のレベルとは逆のレベルになるよう構成すると、リセット時に前記端子の状態でユーザー mode で開始すべきかデバッグモードで開始すべきか判断可能となる。

【0024】

一方外部のデバッグツールが接続されている場合にはデバッグ中であるため、リセット時にはデバッグモードで実行を開始することが好ましい。またデバッグツールが接続されていない場合には、ユーザーが意識しなくとも、ユーザーモードで実行を開始することが好ましい。

【0025】

本発明によれば、前記デバッグ用端子がハイであるかローであるかを検出するという簡単な構成で、ユーザー側では意識することなくリセット時に適切なモードで実行を開始することができる。

【0026】

また本発明の電子機器は、上記いずれかに記載のマイクロコンピュータと、前記マイクロコンピュータの処理対象となるデータの入力源と、前記マイクロコンピュータにより処理されたデータを出力するための出力装置とを含むことを特徴とする。

【0027】

このようにすれば、電子機器を動作させるプログラムなどのデバッグ作業の効率化を図れるようになり、電子機器の開発期間の短縮化、低コスト化を図れるようになる。

【0028】

また本発明は、マイクロコンピュータを含むターゲットシステムのためのデバッグシステムであって、ホストシステムが発行したデバッグコマンドを少なくとも1つのプリミティブコマンドに変換するための処理を行う第2のモニタ手段と

前記第2のモニタ手段との間でデータを送受信し、実行するプリミティブコマンドを前記第2のモニタ手段からの受信データに基づいて決定し、決定したプリミティブコマンドを実行するための処理を行う第1のモニタ手段と、ユーザーモード及びデバッグモードが切り替え可能に形成され、ユーザーモードにおいて前記プリミティブコマンドの実行処理を行う中央処理ユニットと、前記中央処理ユニットを含むチップに設けられ、前記データを半2重の双方向通信を行うための1本の通信ラインが接続されるデバッグ用端子と、前記デバッグ用端子を介して

強制ブレークの入力がされると、前記中央処理ユニットをユーザー モードからデバッガ モードに切り替える切り替え手段とを含むことを特徴とする。

【0029】

本発明によれば、第1のモニタ手段の処理を実行するためのモニタプログラムの命令コードサイズを格段に小さくできる。これにより、ユーザが自由に使用できるメモリ領域や端子数を増やすことが可能になる。また、ターゲットシステムを、実動作時の環境と同一の環境でデバッガできるデバッガシステムを提供できるようになる。

【0030】

【発明の実施の形態】

以下、本発明の好適な実施形態について図面を用いて詳細に説明する。

【0031】

1. 本実施形態の特徴

まず本実施形態の特徴について図2 (A) (B) を用いて説明する。図2 (A) (B) はいずれもユーザー モード及びデバッガ モードを有するマイクロコンピュータ10とデバッガツール14が接続されている様子を示した図である。

【0032】

マイクロコンピュータ10はCPU(中央処理ユニット)12とオンチップデバッガ部13、強制ブレーク制御部15、クロック生成部17を含む。オンチップデバッガ部13はデバッガ モード時にデバッガツール14とデバッガ情報の送受を行い、デバッガ用プログラムを実行して種々のデバッガ処理を行う。

【0033】

また、マイクロコンピュータ10はデバッガ用端子としてSIOD16とBCLK18を含んでいる。SIOD16はユーザー モード時には強制ブレークの入力信号を受け付け、デバッガ モード時には調歩同期式のSIOで外部のデバッガツール14とデバッガ情報の送受信を行う。BCLK18はクロック生成部17により供給され、調歩同期式の同期クロックとして使用される。

【0034】

ところで外部のデバッガツール14との送受信はデバッガ モード時のみに行わ

れ、ユーザー モード 時にはこの デバッ グ用 端子 は 使用され ない。一方 強制 ブレー クの 入力 は ユーザー モード から デバッ グモード へ 切り替えるため の 入力 であるた め ユーザー モード 時のみ 発生 する。

【0035】

従って 強制 ブレー ク制御 部 15 は、 ユーザー モード 時に 信号 を 受けた 場合 には 強制 ブレー クの 入力 が あつた もの と 判断し、 C P U 12 を ユーザー モード から デバッ グモード に 切り替える 处理 を 行う (図 2 (A) 参照) 。 また デバッ グモード 時に 信号 を 受けた 場合 には デバッ グ情報 である 判断し、 当該 デバッ グ情報 を オン チップ デバッ グ部 13 に 出力 する (図 2 (B) 参照) 。

【0036】

この よう に デバッ グ情報 の 送受信 と 強制 ブレー クの 入力 は 異なる モード でしか 発生 しない ため、 図 2 (A) (B) に 示す よう に 同一 の 端子 を 共用 しても それぞれ を 判別 可能 で 混同 は 生じ ない。

【0037】

そこで 本 実施の 形態 では、 S I O D 16 は、 ユーザー モード では 図 2 (A) に 示す よう に 強制 ブレー クの 入力 信号 を 入力 する 端子 として 機能 し、 デバッ グモード では 図 2 (B) に 示す よう に デバッ グ情報 の 送受信 を 行う ため の 端子 として 機能 する よう 構成 さ れ て いる。

【0038】

この よう に す る こ と で デバッ グ 時 に 必要 な 端子 数 を 節約 し、 ユーザー が 使用 可能 な 端子 を より 多く 確保 す る こ と が で きる。

【0039】

図 3 は 強制 ブレー クの 入力 と S I O D 端子 の 状態 の 関係 を 表した タイミング チャート 図 である。

【0040】

16' は デバッ グツール 14 側 の 出力 状態 の S I O D を 示し、 16 は マイクロ コンピュータ 側 の 入力 状態 の S I O D を 示して いる。 デバッ グツール 側 は ユーザー モード 中 は ハイ レベル の 信号 を 出力 して おり、 外部 から の 強制 ブレー クの 入力 により ロー パルス が 出力 さ れる (222) 。 マイクロ コンピュータ 側 では この ロ

一パルスを受けて、ユーザー モードからデバッグ モードに移行する（228）。

そしてデバッグツール14側は、一定期間ハイレベルの信号を出力（224）した後、調歩同期式でデバッグ情報の通信を開始する。

【0041】

図4（A）（B）は本実施の形態の第二の特徴について説明するための図である。図4（A）に示すように、マイクロコンピュータ10がデバッグツール未接続時には、S I O D端子はプルアップのためハイレベルになる。ところが図4（B）に示すようにデバッグツール14を接続することにより、S I O D端子16を任意のレベル（ハイレベル又はローレベル）にすることができる。

【0042】

本実施の形態では、デバッグツール14をマイクロコンピュータ10に接続することにより、S I O D端子16がローレベルになるようデバッグツール14からローレベルの信号を出力している。

【0043】

図5（A）（B）は、S I O D端子の状態とリセット時の開始モードを説明するための図である。

【0044】

図5（A）に示すようにユーザーRESETの立ち上がり（230）でS I O Dがハイレベルの場合には、マイクロコンピュータはユーザー モードで実行を開始する（232）。

【0045】

図5（B）に示すようにユーザーRESETの立ち上がり（234）でS I O Dがローレベルの場合にはマイクロコンピュータはデバッグ モードで実行を開始する（236）。従って、マイクロコンピュータはデバッグ モードに移行し（238）、デバッグツール側は、一定期間ハイレベルの信号を出力（240）した後、調歩同期式でデバッグ情報の通信を開始する（242）。

【0046】

このように、ユーザーリセット時にS I O Dがハイレベルであるかローレベルであるかを検出するという簡単な構成で、ユーザー側では意識することなくリセ

ット時に適切なモードで実行を開始することができる。

【0047】

次に本実施の形態の第3の特徴について説明する。図2 (A) (B) においてオンチップデバッグ部14が以下に説明するミニモニタ部314に相当する機能を有している場合に相当する。

【0048】

図6に示すように、本実施形態では、マイクロコンピュータ10が、CPU(中央処理ユニット)12及びミニモニタ部(第1のモニタ手段)314を含む。また、マイクロコンピュータ10の外部にはメインモニタ部(第2のモニタ手段)316が設けられている。ここでメインモニタ部316は、例えばホストシステムなどが発行したデバッグコマンドをプリミティブコマンドに変換(分解)するための処理を行う。また、ミニモニタ部314は、メインモニタ部316との間でデータを送受信する。そして、ミニモニタ部314は、実行するプリミティブコマンドを、メインモニタ部316からの受信データに基づいて決定し、プリミティブコマンドを実行するための処理を行う。

【0049】

ここで、メインモニタ部316の変換処理の対象となるデバッグコマンドとしては、プログラムロード、GO、ステップ実行、メモリライト、メモリリード、内部レジスタライト、内部レジスタリード、ブレークポイント設定、ブレークポイント解除などのコマンドを考えることができる。メインモニタ部316は、これらの多様で複雑なデバッグコマンドを、例えばGO、ライト(デバッグモード時におけるメモリマップ上の所与のアドレスへのライト)、リード(メモリマップ上の所与のアドレスからのリード)などの、シンプルでプリミティブなコマンドに変換する処理を行う。このようにすることで、ミニモニタ部314の処理を行うミニモニタプログラムの命令コードサイズを格段に小さくすることができる。これにより、マイクロコンピュータ10のオンチップデバッグ機能を実現できるようになる。

【0050】

通常デバッグ用のプログラムはプログラムロード、GO、ステップ実行などの

デバッグコマンドの全ての処理ルーチンを有しているため、大きなメモリを必要としマイクロコンピュータに内蔵することは事実上困難であった。

【0051】

しかし本実施形態では、ミニモニタ部314の処理を行うミニモニタプログラムは、GO、ライト、リードなどのシンプルなプリミティブコマンドの処理ルーチンのみを有し、命令コードサイズが非常に小さい（例えば256バイト）。したがって、ミニモニタプログラムをマイクロコンピュータ10に内蔵することが可能となり、オンチップデバッグ機能を実現できるようになる。また、ユーザが自由に使用できるメモリ領域が減少してしまうのを最小限から0に抑えることも可能になる。

【0052】

2. 詳細な構成例

図7に本実施形態のマイクロコンピュータ及びデバッグシステムの詳細な構成例を示す。図7に示すように、マイクロコンピュータ20は、CPU22、BCU（バス制御ユニット）26、内部メモリ（ミニモニタROM42及びミニモニタRAM44以外の内部ROM及び内部RAM）28、クロック生成部30、ミニモニタ部40（第1のモニタ手段）、トレース部50を含む。

【0053】

ここでCPU22は、種々の命令の実行処理を行うものであり、内部レジスタ24を含む。内部レジスタ24は、汎用レジスタであるR0～R15や、特殊レジスタであるSP（スタックポインタレジスタ）、AHR（積和結果データの上位レジスタ）、ALR（積和結果データの下位レジスタ）などを含む。なお、CPU22は、ユーザーモードとデバッグモードを有しており、ライン51を介して強制ブレーク制御部49からの強制ブレークの入力を受けるとユーザーモードからデバッグモードに切り替わるように構成されている。

【0054】

BCU26はバスを制御するものである。例えば、CPU22に接続されるハーバードアーキテクチャのバス31や、内部メモリ28に接続されるバス32や、外部メモリ36に接続される外部バス33や、ミニモニタ部40、トレース部

50などに接続される内部バス34の制御を行う。またクロック生成部30は、マイクロコンピュータ20内で使用される各種のクロックを生成するものである。クロック生成部30はBCLKを介して外部のデバッグツール60にもクロックを供給している。

【0055】

ミニモニタ部40は、ミニモニタROM42、ミニモニタRAM44、制御レジスタ46、SIO48、強制ブレーク制御部49を含む。

【0056】

ここで、ミニモニタROM42には、ミニモニタプログラムが格納される。本実施形態では、このミニモニタプログラムは、GO、リード、ライトなどのシンプルでプリミティブなコマンドの処理のみを行うようになっている。このため、ミニモニタROM42のメモリ容量を例えば256バイト程度に抑えることができ、オンチップデバッグ機能を持たせながらマイクロコンピュータ20を小規模化できるようになる。

【0057】

ミニモニタRAM44には、デバッグモードへの移行時に（ユーザプログラムのブレーク発生時に）、CPU22の内部レジスタ24の内容が退避される。これにより、デバッグモードの終了後にユーザプログラムの実行を適正に再スタートできるようになる。また内部レジスタの内容のリード等を、ミニモニタプログラムが持つプリミティブなリードコマンド等で実現できるようになる。

【0058】

制御レジスタ46は、各種のデバッグ処理を制御するためのレジスタであり、ステップ実行イネーブルビット、ブレークイネーブルビット、ブレークアドレスビット、トレースイネーブルビットなどを有する。ミニモニタプログラムにより動作するCPU22が制御レジスタ46の各ビットにデータをライトしたり、各ビットのデータをリードすることで、各種のデバッグ処理が実現される。

【0059】

SIO48は、マイクロコンピュータ20の外部に設けられたデバッグツール60との間で送受信するデバッグ用のデータの制御を行うものである。

【0060】

強制ブレーク制御部49とデバッグツール60との間は、S I O D（データ送受信ライン）で接続されており、強制ブレークの入力及びデバッグ情報が送受信される。

【0061】

強制ブレーク制御部49は、前記S I O Dを介してユーザーモード時に信号を受けた場合には強制ブレークの入力があったものと判断し、ライン51を介してC P U 1 2をユーザーモードからデバッグモードに切り替えるための信号を送る処理を行う。また、デバッグモード時に信号を受けた場合にはデバッグ情報であると判断し、当該デバッグ情報をS I O 4 8に出力する処理を行う。

【0062】

トレース部50は、リアルタイムトレース機能を実現するためのものである。トレース部50とデバッグツール60との間は、C P U 2 2の命令実行のステートを表す3ビットのD S T [2:0]と、分岐先のP C（プログラムカウンタ）値を表すD P C Oという4本のラインで接続されている。

【0063】

デバッグツール60はメインモニタ部62を含み、パーソナルコンピュータ等により実現されるホストシステム66に接続される。ホストシステム66が、ユーザの操作により、プログラムロード、ステップ実行などのデバッグコマンドを発行すると、メインモニタ部62が、このデバッグコマンドをプリミティブコマンドに変換（分解）するための処理を行う。そして、メインモニタ部62が、プリミティブコマンドの実行を指示するデータをミニモニタ部40に送信すると、ミニモニタ部40が、指示されたプリミティブコマンドを実行するための処理を行うことになる。

【0064】

図8に、デバッグモード時のメモリマップの例を示す。図8のD1、D2、D3に示すように、デバッグモード時には、図7の制御レジスタ46、ミニモニタRAM44、ミニモニタROM42のアドレスも、メモリマップ上に割り付けられる。

【0065】

3. プリミティブコマンドへの変換

図9 (A)、(B)、(C)、(D)に、各種のデバッグコマンドをプリミティブコマンドへ変換する処理について模式的に示す。

【0066】

例えば図9 (A)に示すように、(ADD…、SUB…、AND…、OR…、XOR…、LD. W…) という12バイトのプログラムを80010h番地にロードするというデバッグコマンドが発行されたとする。この場合、このプログラムロードコマンドは、ライト(80010h, ADD…, SUB)、ライト(80014h, AND…, OR…)、ライト(80018h, XOR…, LD. W…) という3つのプリミティブなライトコマンドに変換される。即ち、ミニモニタプログラムが、この3つのプリミティブなライトコマンドを実行することで、プログラムロードコマンドが実現されるようになる。

【0067】

また図9 (B)に示すようにステップ実行コマンドというデバッグコマンドが発行されたとする。すると、このステップ実行コマンドは、図7の制御レジスタ46のステップ実行イネーブルビットへのライトコマンド(図8のD1のアドレスへのライトコマンド)とGOコマンドに変換される。即ち、ミニモニタプログラムが、このプリミティブなライトコマンドとGOコマンドを実行することで、ステップ実行コマンドが実現されるようになる。

【0068】

また図9 (C)に示すように内部レジスタリードコマンドというデバッグコマンドが発行されたとする。すると、この内部レジスタリードコマンドは、メモリマップ上のミニモニタRAM(内部レジスタの内容の退避先)からのリードコマンド(図8のD2のアドレスからのリードコマンド)に変換される。即ち、ミニモニタプログラムが、このプリミティブなリードコマンドを実行することで、内部レジスタリードコマンドが実現されるようになる。内部レジスタライトコマンド、メモリリードコマンド、メモリライトコマンドも同様にして実現される。

【0069】

また図9 (D) に示すようにブレークポイント設定コマンドというデバッグコマンドが発行されたとする。すると、このブレークポイント設定コマンドは、制御レジスタ46のブレークイネーブルビット及びブレークアドレスビットへのライトコマンドに変換される。即ち、ミニモニタプログラムが、このプリミティブなライトコマンドを実行することで、ブレークポイント設定コマンドが実現されるようになる。

【0070】

このように本実施形態では、複雑で多様なデバッグコマンドが、プリミティブでシンプルなリード、ライト、GOコマンドに変換される。そして、ミニモニタプログラムは、このプリミティブなリード、ライト、GOコマンドを実行するだけでよいため、ミニモニタプログラムの命令コードサイズは非常に小さくなる。この結果、ミニモニタROM42のメモリ容量も小さくでき、小さなハードウェア規模でオンチップデバッグ機能を実現できるようになる。

【0071】

4. S I Oの構成例

図10にS I O 4 8の構成例を示す。S I O 4 8は、送受信バッファ70、シフトレジスタ76、送受信切替部78、クロック制御部80及び制御レジスタ84を含む。

【0072】

ここで送受信バッファ70は、送信データ、受信データを一時的に蓄えるためのものであり、送信バッファ72、受信バッファ74を有する。シフトレジスタ76は、送信バッファ72からの送信データをパラレルデータからシリアルデータに変換し送受信切替部78に出力する機能を有する。また送受信切替部78からの受信データをシリアルデータからパラレルデータに変換し受信バッファ74に出力する機能も有する。送受信切替部78は、データの送信と受信とを切り替えるためのものである。これにより、S I O Dを使用した半2重のデータ送受信が可能になる。

【0073】

クロック制御部80は、内蔵する分周回路82によりB C L Kを分周し、この

分周により得られたサンプリングクロック SMC 1 をシフトレジスタ 76 に出力する。シフトレジスタ 76 は、この SMC 1 に基づき動作する。またこの BCLK はデバッグツール 60 に供給されるため、マイクロコンピュータ 20 とデバッゲツール 60 により、BCLK が共有されるようになる。

【0074】

分周回路 82 での分周比は制御レジスタ 84 により設定される。即ち CPU 22 により実行されるミニモニタプログラムが、所望の分周比を制御レジスタ 84 に書き込むことで、分周回路 82 での分周比が設定されることになる。

【0075】

5. デバッグツールの構成例

図 11 にデバッグツール 60 の構成例を示す。

【0076】

CPU 90 は、ROM 108 に格納されるプログラムを実行したり、デバッグツール 60 の全体の制御を行うものである。送受信切替部 92 は、データの送信と受信とを切り替えるためのものである。クロック制御部 94 は、CPU 90 の SCLK 端子、アドレスアップカウンタ 100、トレースメモリ 104 に供給するクロックを制御するものである。このクロック制御部 94 には、マイクロコンピュータ 20 (SIO 48) からの BCLK が入力される。クロック制御部 94 は周波数検出回路 95、分周回路 96 を含む。周波数検出回路 95 は、BCLK が属する周波数範囲を検出して、その結果を制御レジスタ 98 に出力する。また分周回路 96 での分周比は制御レジスタ 98 により制御される。即ち CPU 90 により実行されるメインモニタプログラム (メインモニタ ROM 110 に格納) が、制御レジスタ 98 から BCLK の周波数範囲を読み出す。そして、メインモニタプログラムは、この周波数範囲に応じた最適な分周比を決定し、この分周比を制御レジスタ 98 に書き込む。そして、分周回路 96 は、この分周比で BCLK を分周して SMC 2 を生成し、CPU 90 の SCLK 端子に出力する。

【0077】

アドレスアップカウンタ 100 は、トレースメモリのアドレスをカウントアップするためのものである。セレクタ 102 は、ライン 122 (アドレスアップカ

ウンタ100が出力するアドレス)とライン124(アドレスバス120からのアドレス)のいずれかを選択し、トレースメモリ104のアドレス端子にデータを出力する。またセレクタ106は、ライン126(図3のトレース部50の出力であるDST[2:0]、DPCO)とライン128(データバス118)のいずれかを選択し、トレースメモリ104のデータ端子にデータを出力したり、データ端子からデータを取り出す。

【0078】

ROM108はメインモニタROM110(図3のメインモニタ部62に相当)を含み、メインモニタROM110には、メインモニタプログラムが格納される。このメインモニタプログラムは、図5(A)～図5(D)で説明したように、デバッグコマンドをプリミティブコマンドに変換するための処理を行う。RAM112は、CPU90のワーク領域となるものである。

【0079】

RS232Cインターフェース114、パラレルインターフェース116は、図3のホストシステム66とのインターフェースとなるものであり、ホストシステム66からのデバッグコマンドはこれらのインターフェースを介してCPU90に入力されることになる。クロック生成部118は、CPU90を動作させるクロックなどを生成するものである。

【0080】

6. データの送受信

さて、図12(A)に示すように、ミニモニタ部40とメインモニタ部62の間でのデバッグデータの通信の手法としては、TXD(送信)とRXD(受信)のラインを別々に設け、全2重で通信する手法が考えられる。

【0081】

しかしながら、このようにデバッグデータの通信のために2本のライン(端子)を使用してしまうと、その分だけマイクロコンピュータの端子数(ピン数)が増えてしまい、これはマイクロコンピュータの高コスト化を招く。

【0082】

そこで本実施形態では、図12(B)に示すように、ミニモニタ部40とメイ

ンモニタ部62との間に1本のTxD/RxDライン（双方向通信ライン）を設け、半2重の双方向通信を行う。なお本実施の形態では、このラインを強制ブレークを入力するためのSIODと共用している。このため、マイクロコンピュータの端子数の増加を最小限に抑えることができ、マイクロコンピュータの低コスト化を図れる。

【0083】

そして更に本実施形態では、図12(C)に示すように、スレーブとなるミニモニタ部40が、マスタとなるメインモニタ部62からデータを受信したことを条件に、その受信データに対応する処理を行い、その受信データに対応する応答データをメインモニタ部62に送信する。即ち、メインモニタ部62がデータ（コマンド）をミニモニタ部40に送信すると、ウェイト状態となっていたミニモニタ部40がこれを受信し、その受信データに対応する処理を行う。そして、その受信データに対応するデータ（リプライ）をメインモニタ部62に送信する。その後、ミニモニタ部40は、メインモニタ部62からデータを受信するまでウェイト状態になる。即ちミニモニタ部40は、メインモニタ部62からデータを受信するまで動作を停止し、データを受信したことを条件に動作を開始する。このようにすることで、ミニモニタ部40とメインモニタ部62との間の通信ラインを1本にしながら、データを適正に送受信できるようになる。

【0084】

7. ミニモニタ部の詳細な処理例

次にミニモニタ部の詳細な処理例について説明する。

【0085】

図13に示すように、ユーザプログラムの実行中にブレークが発生すると、ミニモニタプログラムの処理がスタートし、ユーザプログラム実行モードからデバッグモードに移行する。そして、ミニモニタプログラムが所与のコマンド処理を行いリターン命令を実行すると、デバッグモードからユーザプログラム実行モードに戻る。

【0086】

図14、図15に、デバッグモードでのミニモニタプログラムの処理を表すフ

ローチャートを示す。

【0087】

デバッグモードに移行すると、ミニモニタプログラムは、まず、図7のCPU 22の内部レジスタ24の内容をモニタRAM44に退避する（ステップS1）。そして、ミニモニタプログラムが使用する制御レジスタ46の設定処理を行う（ステップS2）。

【0088】

次に、デバッグツール60から受信した14バイトのデータを、受信バッファ74（図10参照）にライトする（ステップS3）。そして、受信バッファ74のデータの先頭の1バイト（コマンド識別データID）をチェックする（ステップS4）。

【0089】

そして図13に示すように、IDがリードコマンドを示すものであった場合には、受信バッファ74からリードアドレスを取得する（ステップS5、S6）。そして、取得したリードアドレスからデータをリードし、送信バッファ72にライトする（ステップS7）。次に、送信バッファ72のデータをデバッグツール60に送信する（ステップS8）。そして、図23のステップS3に戻り、次の受信データを受信バッファ74にライトする。

【0090】

IDがライトコマンドを示すものであった場合には、受信バッファ74からライトアドレスを取得する（ステップS9、S10）。そして、受信バッファ74からライトデータを取得し、ステップS10で取得したライトアドレスにライトする（ステップS11）。

【0091】

IDが外部ルーチンジャンプコマンドを示すものであった場合には、受信バッファ74からルーチンアドレスを取得する（ステップS12、S13）。そして、外部ルーチンにジャンプ後、ミニモニタプログラムにリターンする（ステップS14）。

【0092】

IDがGOコマンドを示すものであった場合には、モニタRAM44に退避したデータを内部レジスタ24にリストアする（ステップS15、S16）。そして、図13に示すようにユーザプログラムにリターンし、デバッグモードから抜ける（ステップS17）。

【0093】

一方、IDがリード、ライト、外部ルーチンジャンプ、GOコマンドのいずれでもなかった場合には、処理が必要ないと判断する（ステップS15、S18）。そして送信バッファ72にダミーのデータをライトする（ステップS19）。なお、図15では、データフィルコマンドの処理については省略している。

【0094】

以上のようにして、デバッグコマンドを変換することで得られたプリミティブコマンドが、ミニモニタプログラムにより実行されることになる。

【0095】

8. 電子機器

次に、以上の本実施形態のマイクロコンピュータを含む電子機器について説明する。

【0096】

例えば図16（A）に電子機器の1つであるカーナビゲーションシステムの内部ブロック図を示し、図17（A）にその外観図を示す。カーナビゲーションシステムの操作はリモコン510を用いて行われ、GPSやジャイロからの情報に基づいて位置検出部520が車の位置を検出する。地図などの情報はCDROM530（情報記憶媒体）に格納されている。画像メモリ540は画像処理の際の作業領域になるメモリであり、生成された画像は画像出力部550を用いて運転者に表示される。マイクロコンピュータ500は、リモコン510、位置検出部520、CDROM530などのデータ入力源からデータを入力し、種々の処理を行い、処理後のデータを画像出力部550などの出力装置を用いて出力する。

【0097】

図16（B）に電子機器の1つであるゲーム装置の内部ブロック図を示し、図17（B）にその外観図を示す。このゲーム装置では、ゲームコントローラ56

0からのプレーヤの操作情報、CD-ROM570からのゲームプログラム、ICカード580からのプレーヤ情報等に基づいて、画像メモリ590を作業領域としてゲーム画像やゲーム音を生成し、画像出力部610、音出力部600を用いて出力する。

【0098】

図16 (C) に電子機器の1つであるプリンタの内部ブロック図を示し、図17 (C) にその外観図を示す。このプリンタでは、操作パネル620からの操作情報、コードメモリ630及びフォントメモリ640から文字情報に基づいて、ピットマップメモリ650を作業領域として、印刷画像を生成し、プリント出力部660を用いて出力する。またプリンタの状態やモードを表示パネル670を用いてユーザに伝える。

【0099】

本実施形態のマイクロコンピュータ又はデバッグシステムによれば、図16 (A) ~図17 (C) の電子機器を動作させるユーザプログラムの開発の容易化、開発期間の短縮化を図れるようになる。またマイクロコンピュータが実動作する環境と同じ環境で、ユーザプログラムのデバッグ作業を行うことができるため、電子機器の信頼性を高めることができる。また電子機器に組み込まれるマイクロコンピュータのハードウェアを小規模化、低コスト化できるため、電子機器の低コスト化も図れるようになる。更にミニモニタプログラムの命令コードサイズは小さいため、ユーザがプログラムや各種データの格納に使用するメモリ領域を全く使用しないことが可能になる。

【0100】

なお本実施形態のマイクロコンピュータを適用できる電子機器としては、上記以外にも例えば、携帯電話（セルラーフォン）、PHS、ページヤ、オーディオ機器、電子手帳、電子卓上計算機、POS端末、タッチパネルを備えた装置、プロジェクタ、ワードプロセッサ、パーソナルコンピュータ、テレビ、ビューファインダ型又はモニタ直視型のビデオテープレコーダなど種々のものを考えることができる。

【0101】

なお、本発明は本実施形態に限定されず、本発明の要旨の範囲内で種々の変形実施が可能である。

【0102】

【図面の簡単な説明】

【図1】

CPU置き換え型のICEの例を示す図である。

【図2】

図2 (A) (B) は、本実施形態の特徴について説明するための図である。

【図3】

強制ブレーキの入力とSIO端子の状態の関係を表したタイミングチャート図である。

【図4】

図4 (A) (B) は本実施の形態の第二の特徴について説明するための図である。

【図5】

図5 (A) (B) は、SIO端子の状態とリセット時の開始モードを説明するための図である。

【図6】

本実施の形態の第三の特徴について説明するための図である。

【図7】

本実施形態のマイクロコンピュータ、デバッグシステムの構成例を示す機能ブロック図である。

【図8】

デバッグモード時のメモリマップを示す図である。

【図9】

図9 (A)、(B)、(C)、(D) は、デバッグコマンドをプリミティブコマンドへ変換(分解)する処理について説明するための図である。

【図10】

SIOの構成例を示す機能ブロック図である。

【図11】

デバッグツールの構成例を示す機能ブロック図である。

【図12】

図12 (A)、(B)、(C) は、ミニモニタ部とメインモニタ部との間の通信手法について説明するための図である。

【図13】

ユーザプログラム実行モードからデバッグモードへの移行について説明するための図である。

【図14】

本実施形態の詳細な処理例を説明するためのフローチャートである。

【図15】

本実施形態の詳細な処理例を説明するためのフローチャートである。

【図16】

図16 (A)、(B)、(C) は、種々の電子機器の内部ブロック図の例である。

【図17】

図17 (A)、(B)、(C) は、種々の電子機器の外観図の例である。

【符号の説明】

| | |
|-----|------------------|
| 1 0 | マイクロコンピュータ |
| 1 2 | C P U (中央処理ユニット) |
| 1 3 | オンチップデバッグ部 |
| 1 4 | デバッグツール |
| 1 5 | 強制ブレーク制御部 |
| 1 6 | S I O D |
| 1 7 | クロック制御部 |
| 1 8 | B C L K |
| 2 0 | マイクロコンピュータ |
| 2 2 | C P U (中央処理ユニット) |
| 2 4 | 内部レジスタ |

2 6 B C U (バス制御ユニット)
2 8 内部メモリ
3 0 クロック生成部

3 1 バス
3 2 バス
3 3 外部バス
3 4 内部バス
3 6 外部メモリ
4 0 ミニモニタ部 (第1のモニタ手段)
4 2 ミニモニタROM
4 4 ミニモニタRAM
4 6 制御レジスタ
4 8 S I O
4 9 強制ブレーク制御部
5 0 トレース部
6 0 デバッグツール
6 2 メインモニタ部 (第2のモニタ手段)
6 6 ホストシステム
7 0 送受信バッファ
7 2 送信バッファ
7 4 受信バッファ

7 6 シフトレジスタ
7 8 送受信切替部
8 0 クロック制御部
8 2 分周回路
8 4 制御レジスタ
9 0 C P U
9 2 送受信切替部
9 4 クロック制御部

95 周波数検出部

96 分周回路

98 制御レジスタ

100 アドレスアップカウンタ

102 セレクタ

104 トレースメモリ

106 セレクタ

108 ROM

110 メインモニタROM

112 RAM

114 RS232Cインターフェース

116 パラレルインターフェース

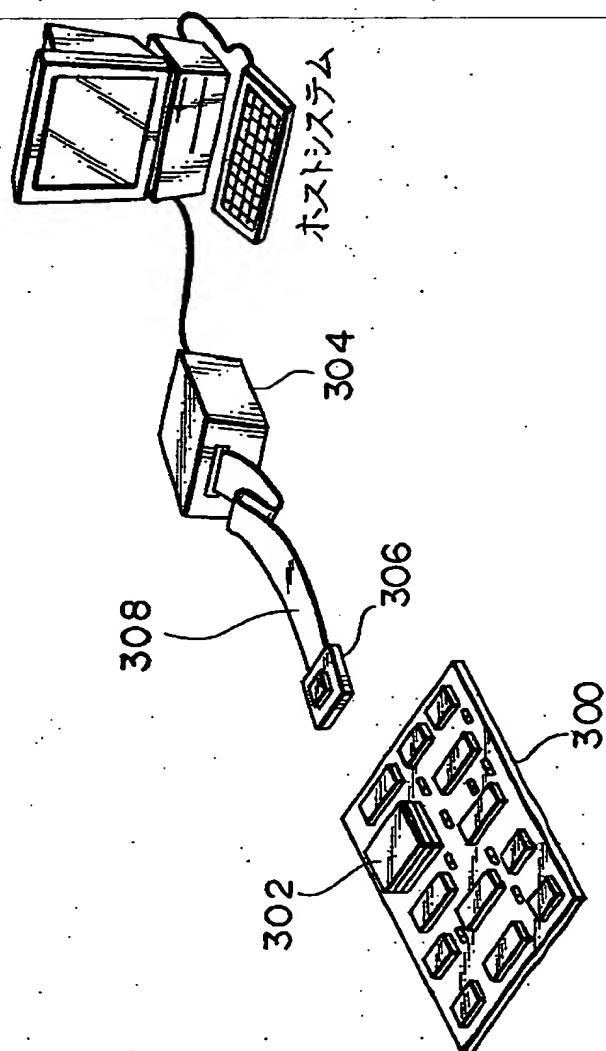
118 クロック生成部

316 メインモニタ部（第2のモニタ手段）

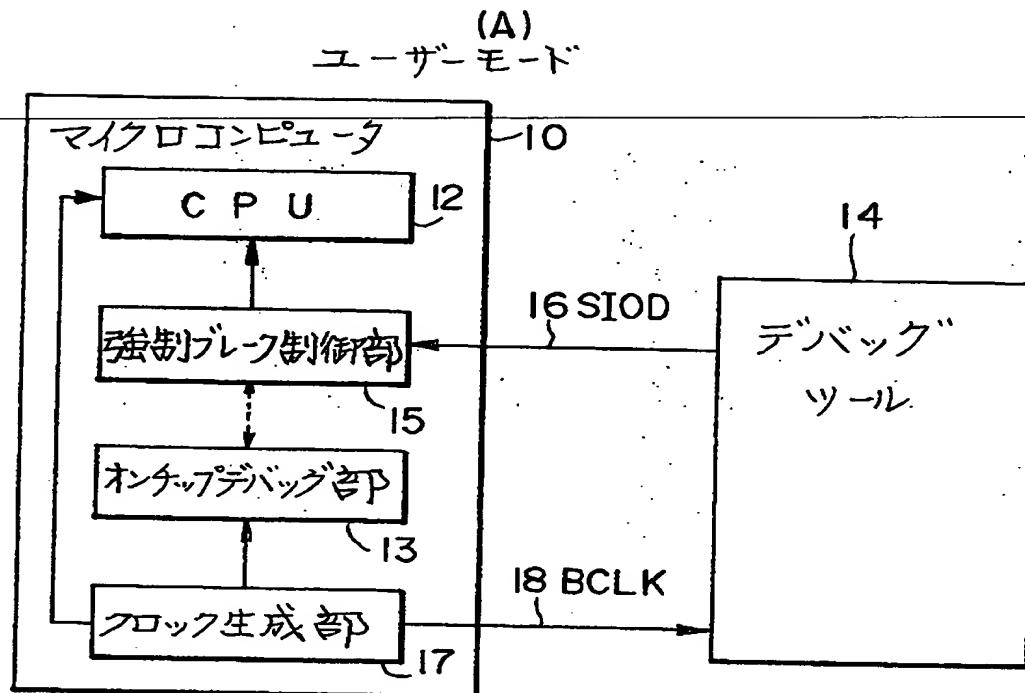
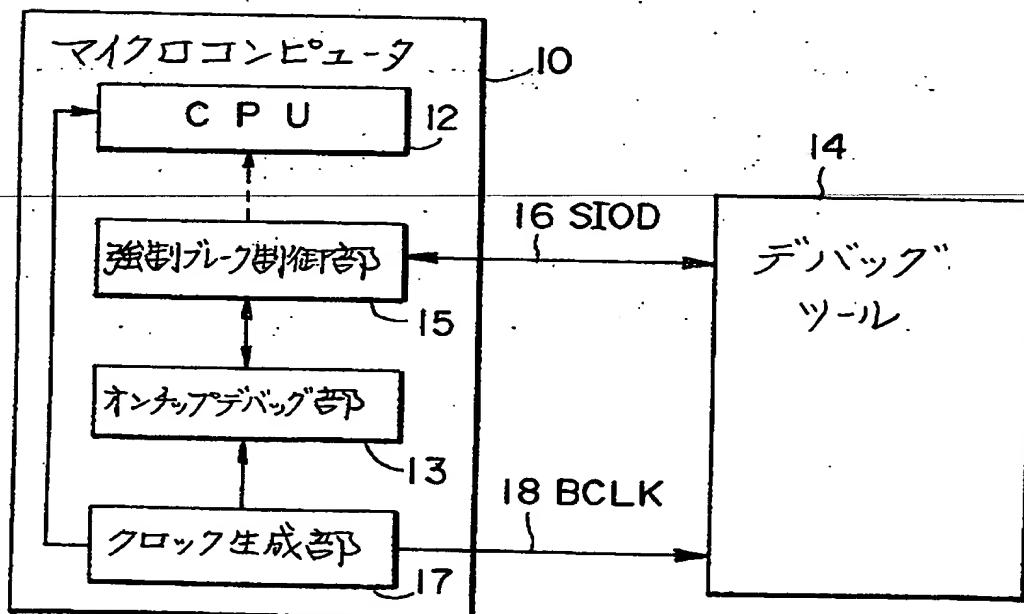
【書類名】

図面

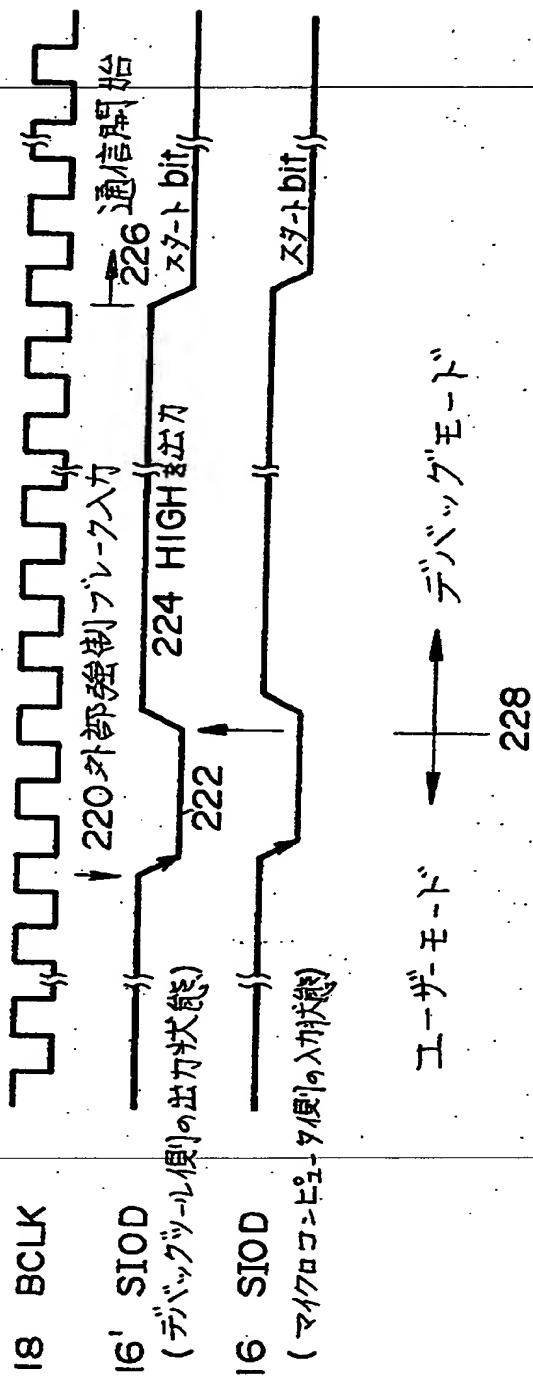
【図1】



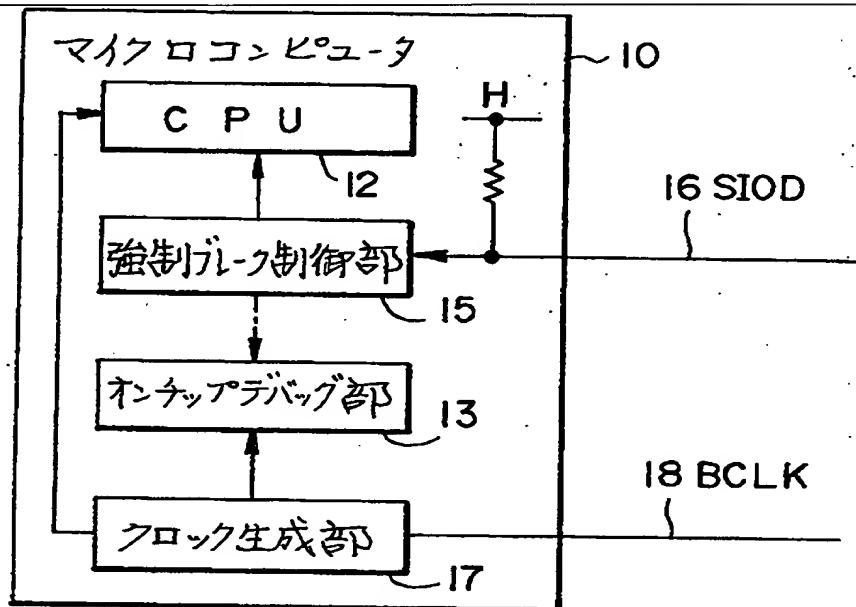
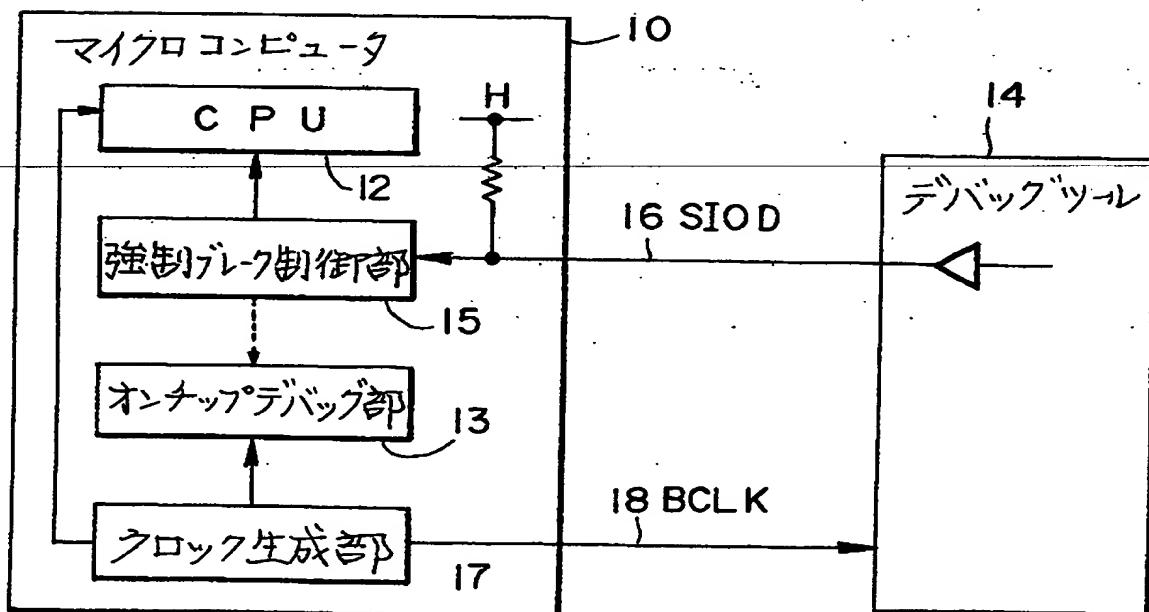
【図2】

(B)
デバッグ モード

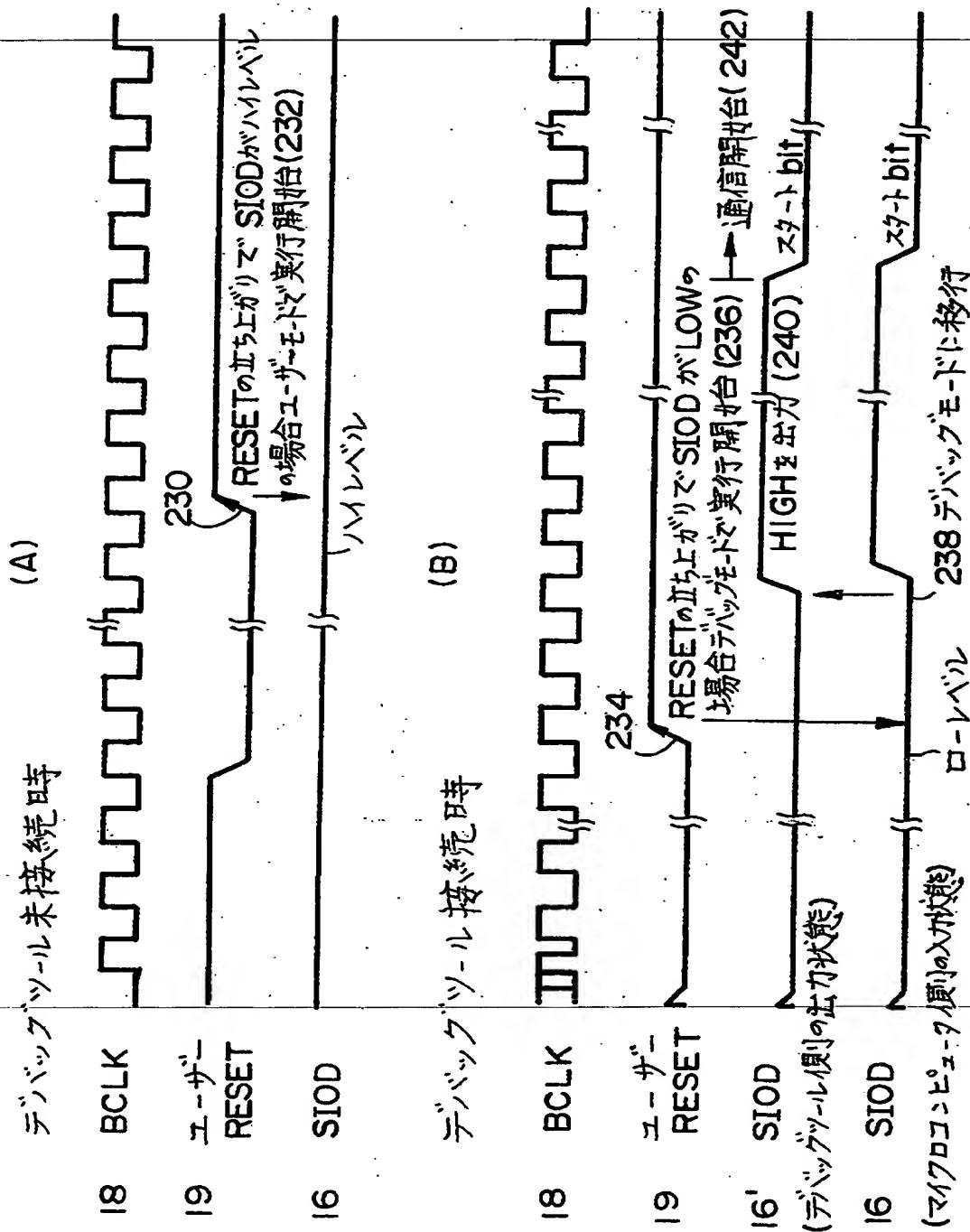
【図3】



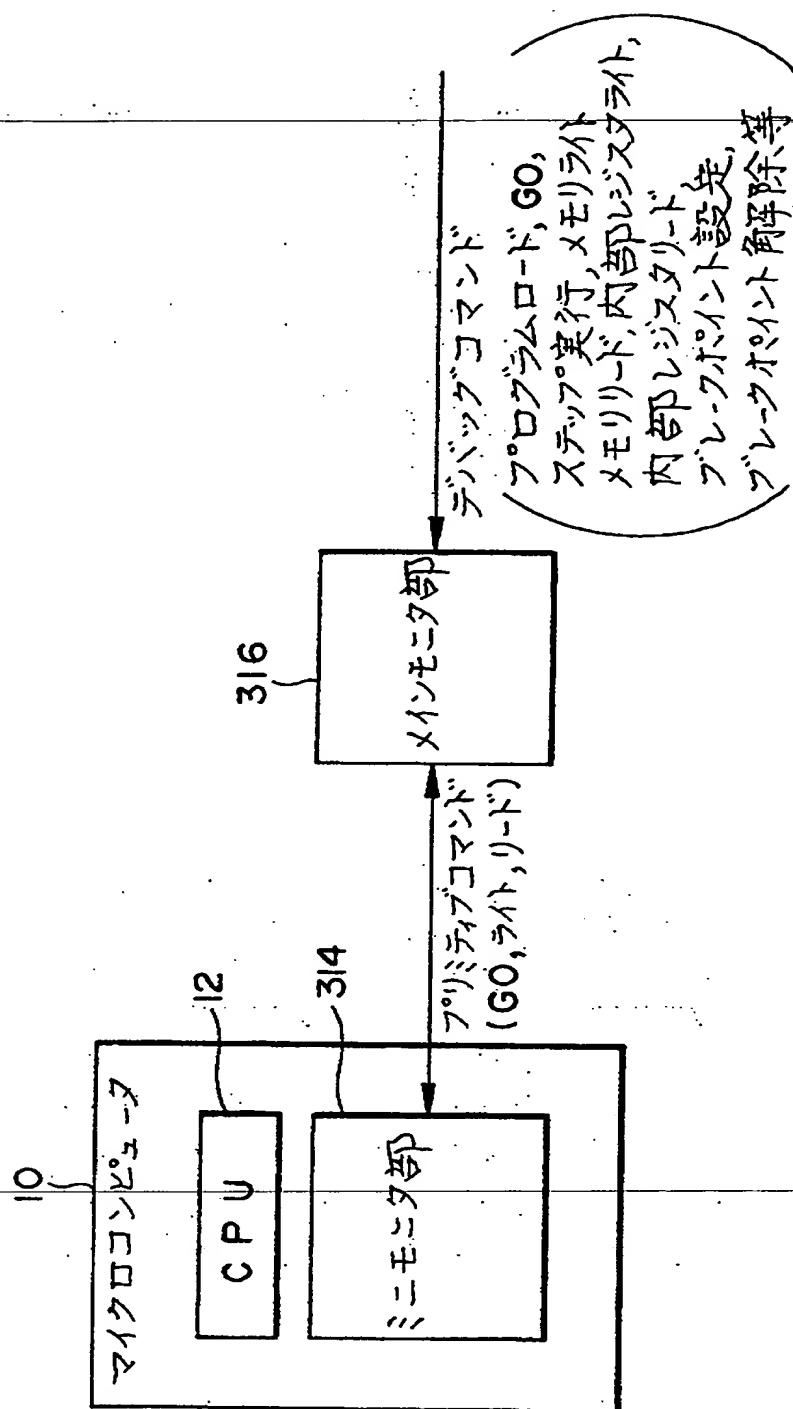
【図4】

(A)
デバッグツール未接続時(B)
デバッグツール接続時

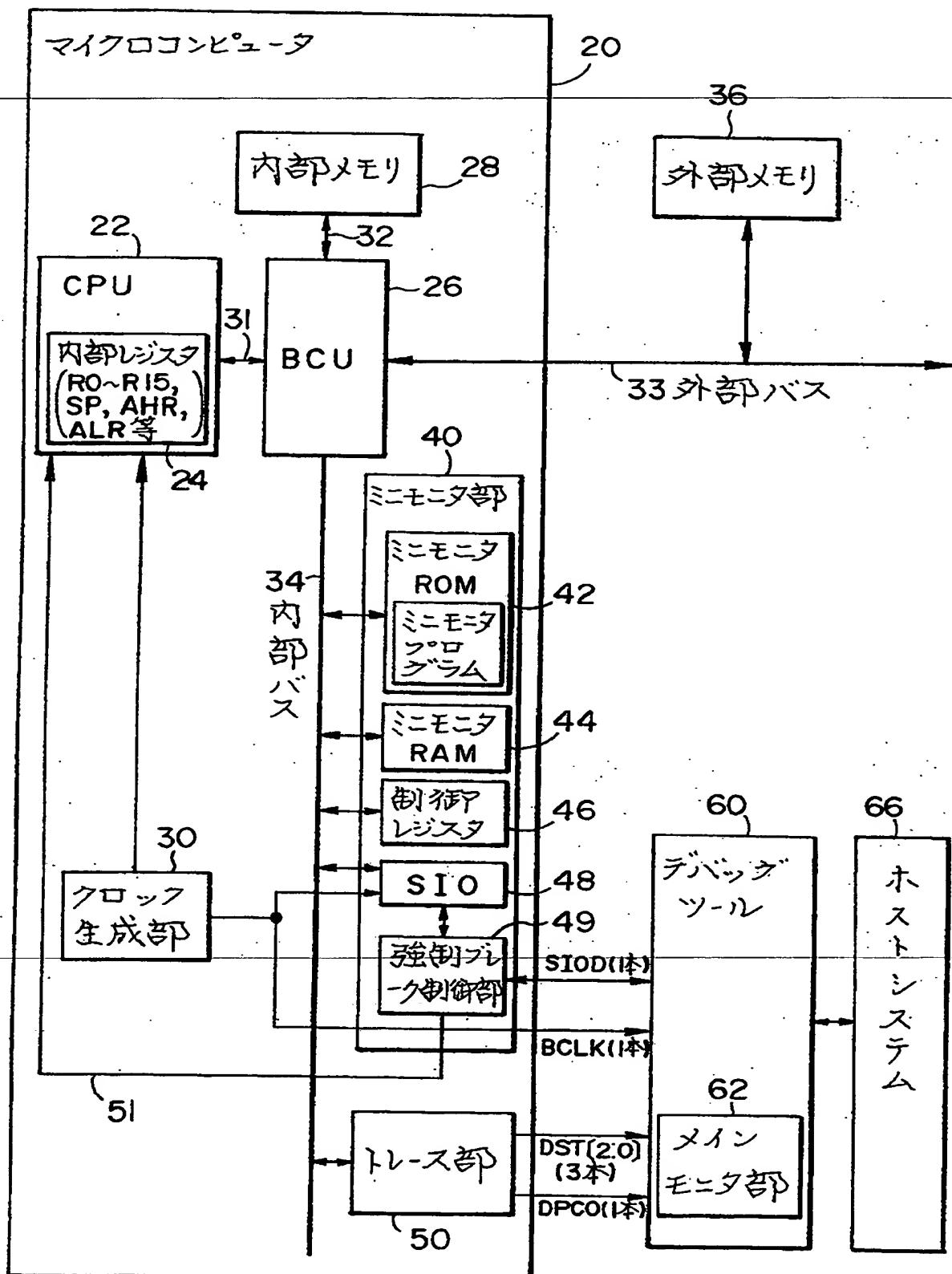
【図5】



【図6】

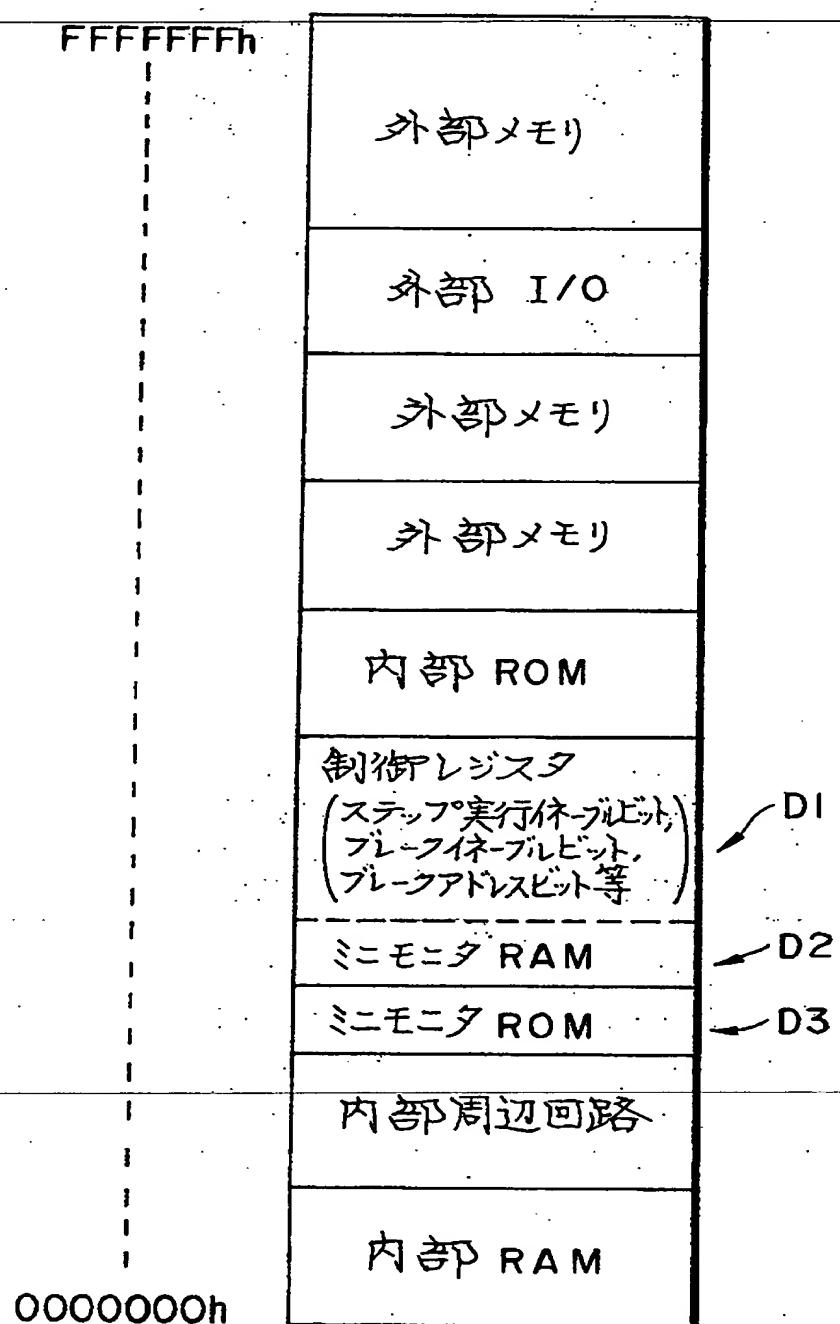


[図7]



【図8】

デバッグモード時のメモリマップ



【図9】

(A) プログラムロード

$$\begin{pmatrix} 80010h, 12バイト, ADD---, SUB--- \\ AND---, OR---, XOR---, LD.W--- \end{pmatrix} \xrightarrow{\quad} \begin{pmatrix} ライト(80010h, ADD---, SUB---) \\ +ライト(80014h, ADD---, OR---) \\ +ライト(80018h, XOR---, LD.W---) \end{pmatrix}$$

(B) ステップ実行

$$\xrightarrow{\quad} \begin{pmatrix} 制御レジスタのステップ実行イネーブル \\ ビットへのライト \\ +GO \end{pmatrix}$$

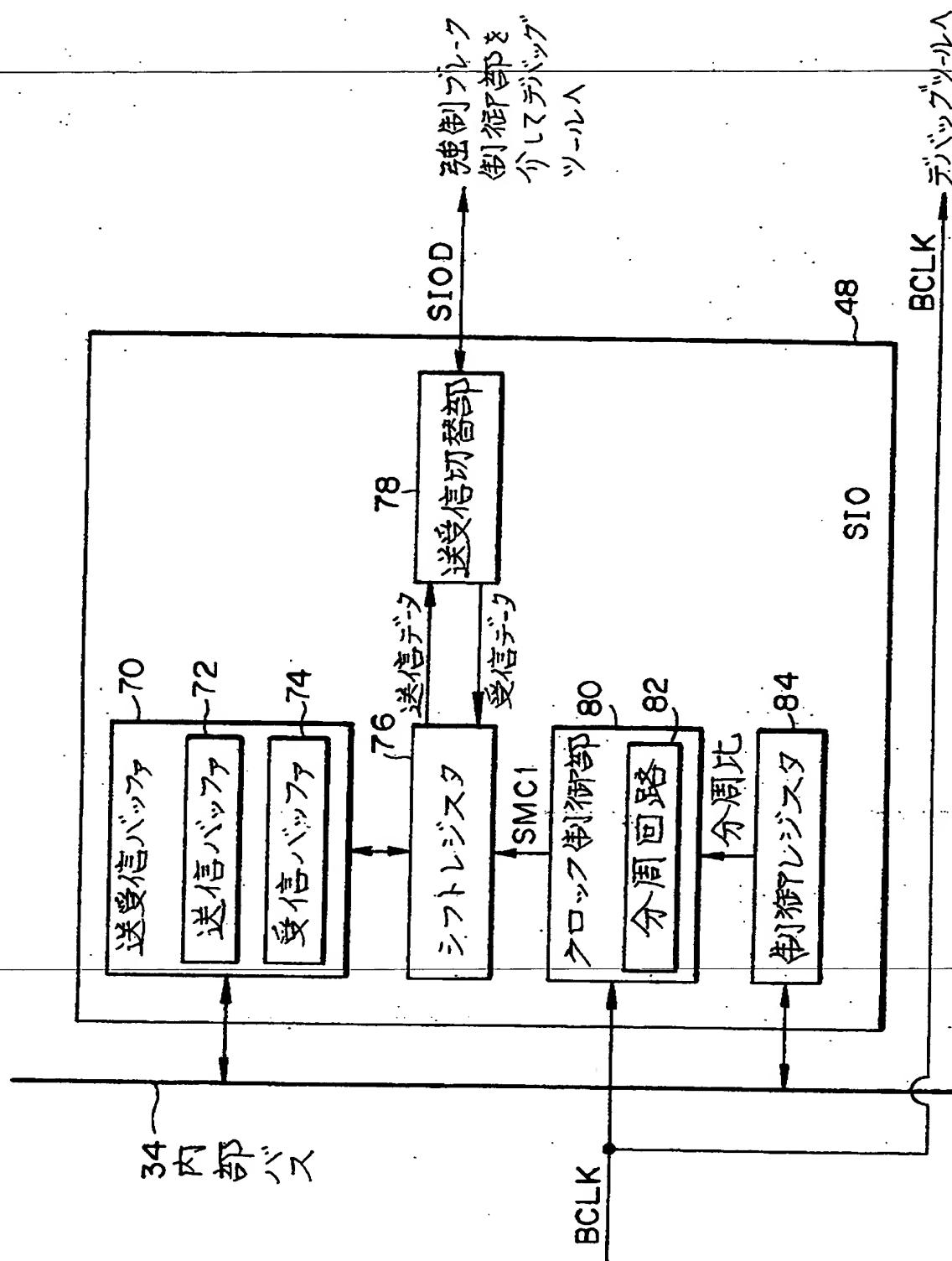
(C) 内部レジスタリード

$$\xrightarrow{\quad} メモリマップ上のモニタRAMへのリード$$

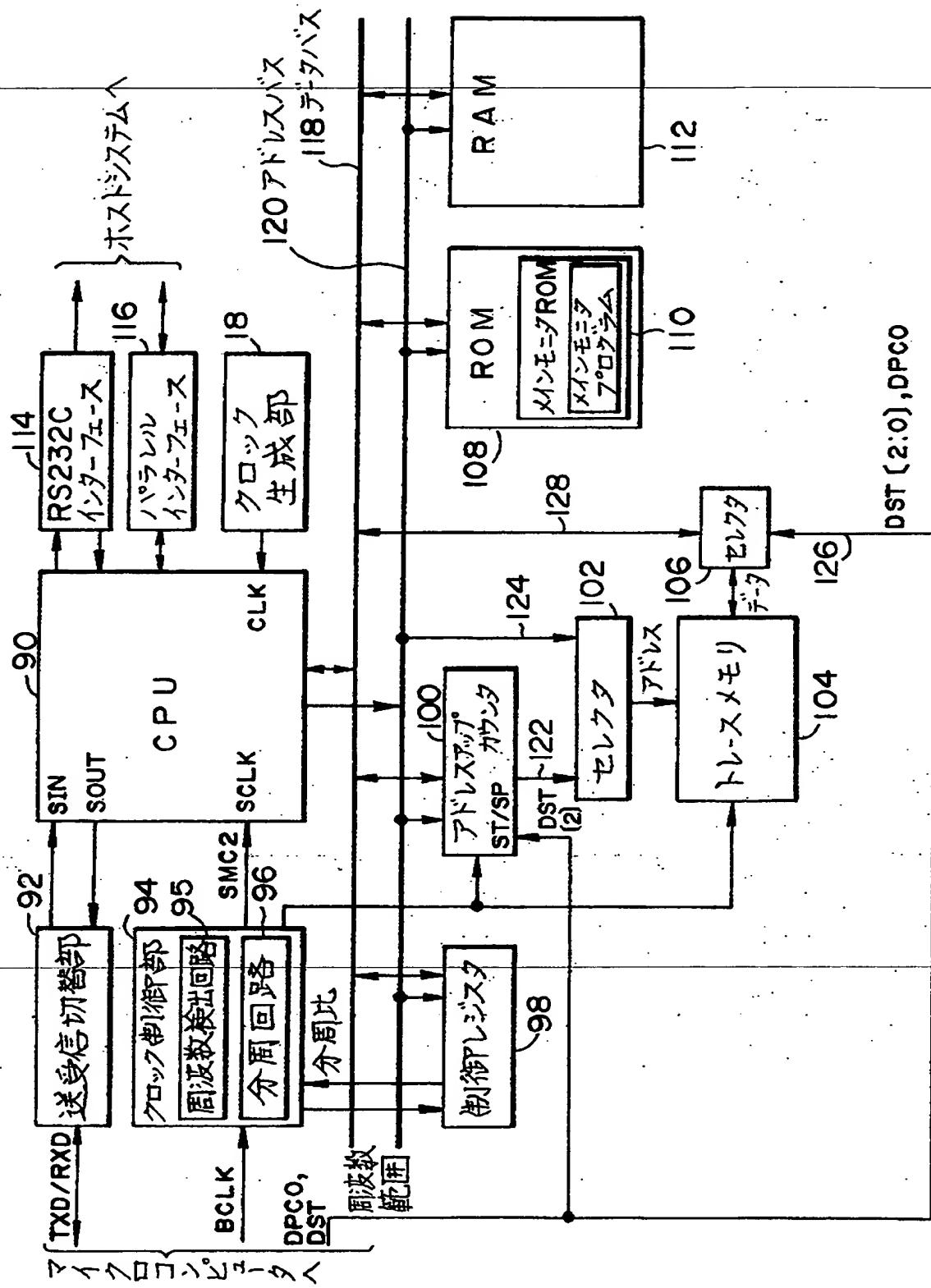
(D) ブレークポイント設定

$$\xrightarrow{\quad} \begin{pmatrix} 制御レジスタのブレークイネーブルビット \\ 及びブレークアドレスビットへのライト \end{pmatrix}$$

【図10】

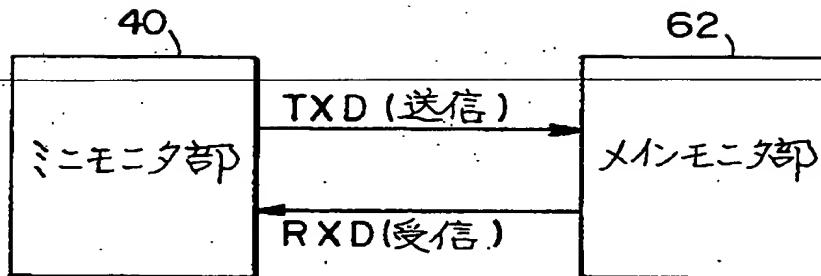


【図11】

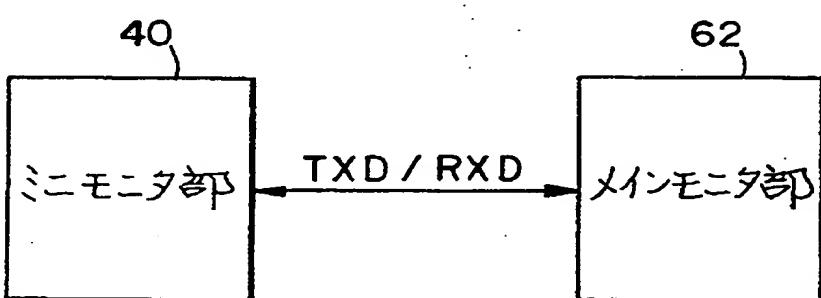


【図12】

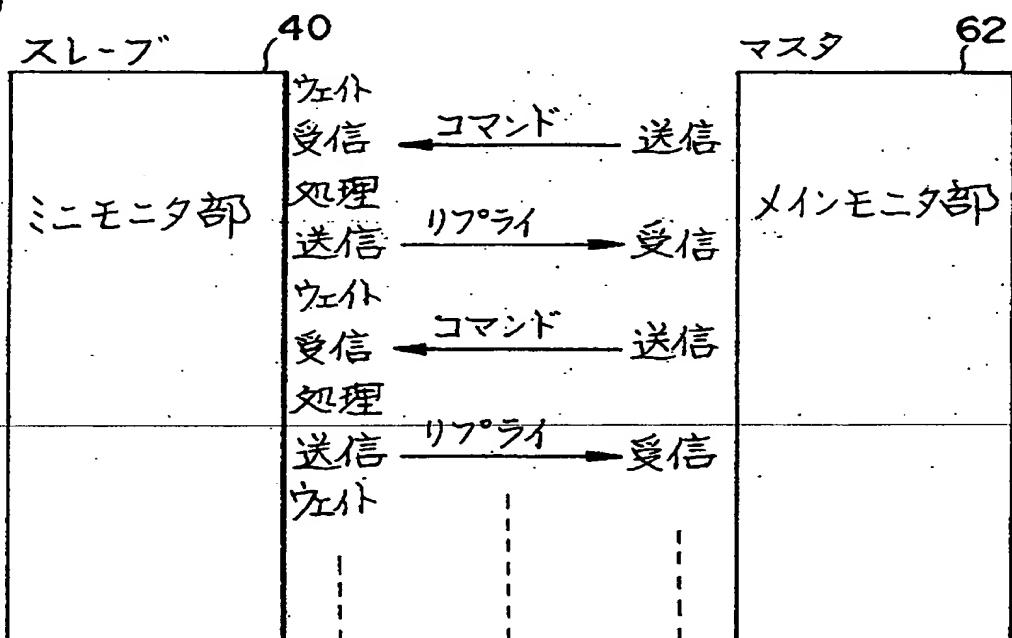
(A)



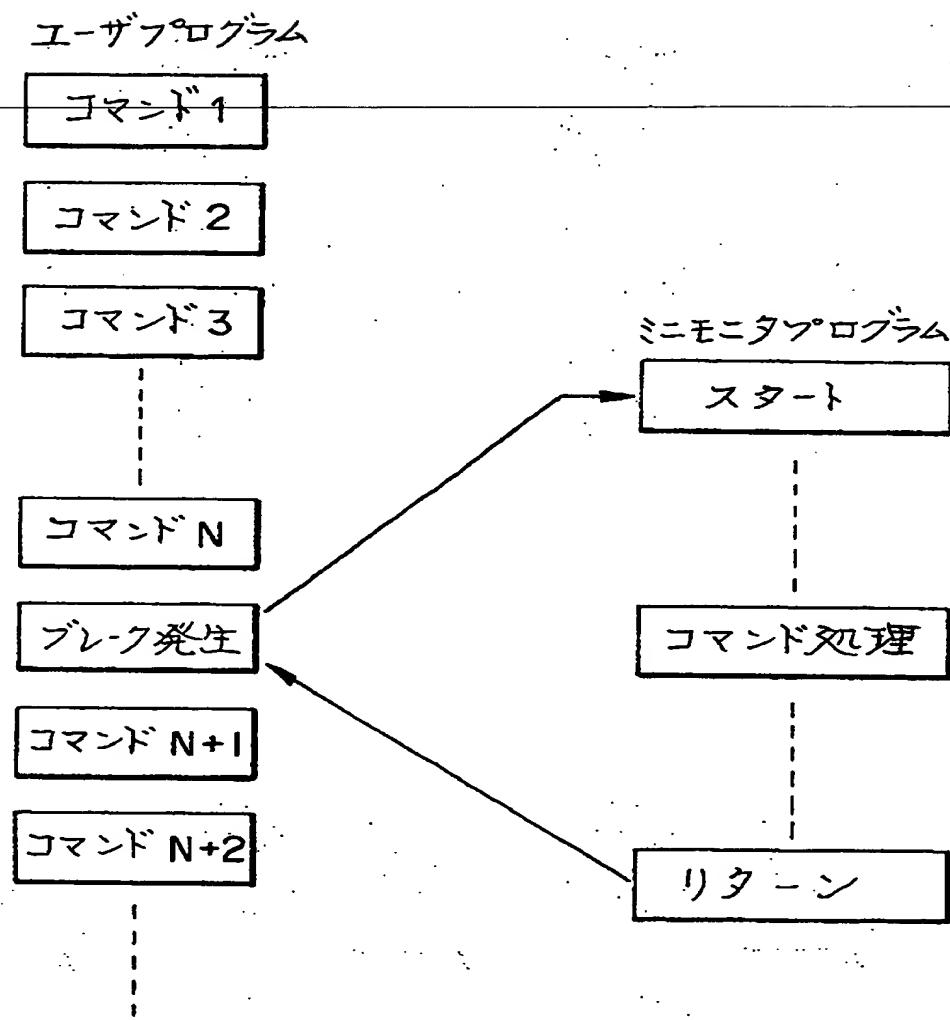
(B)



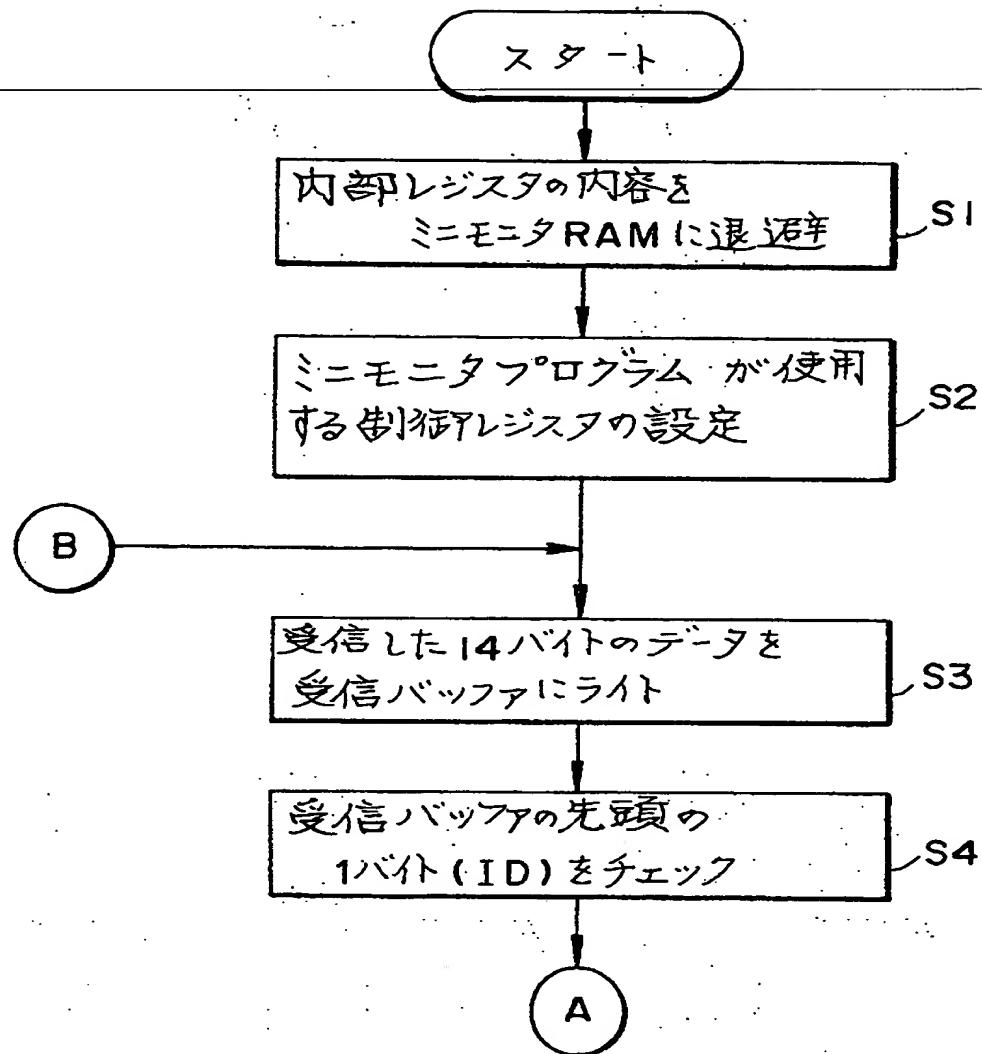
(C)



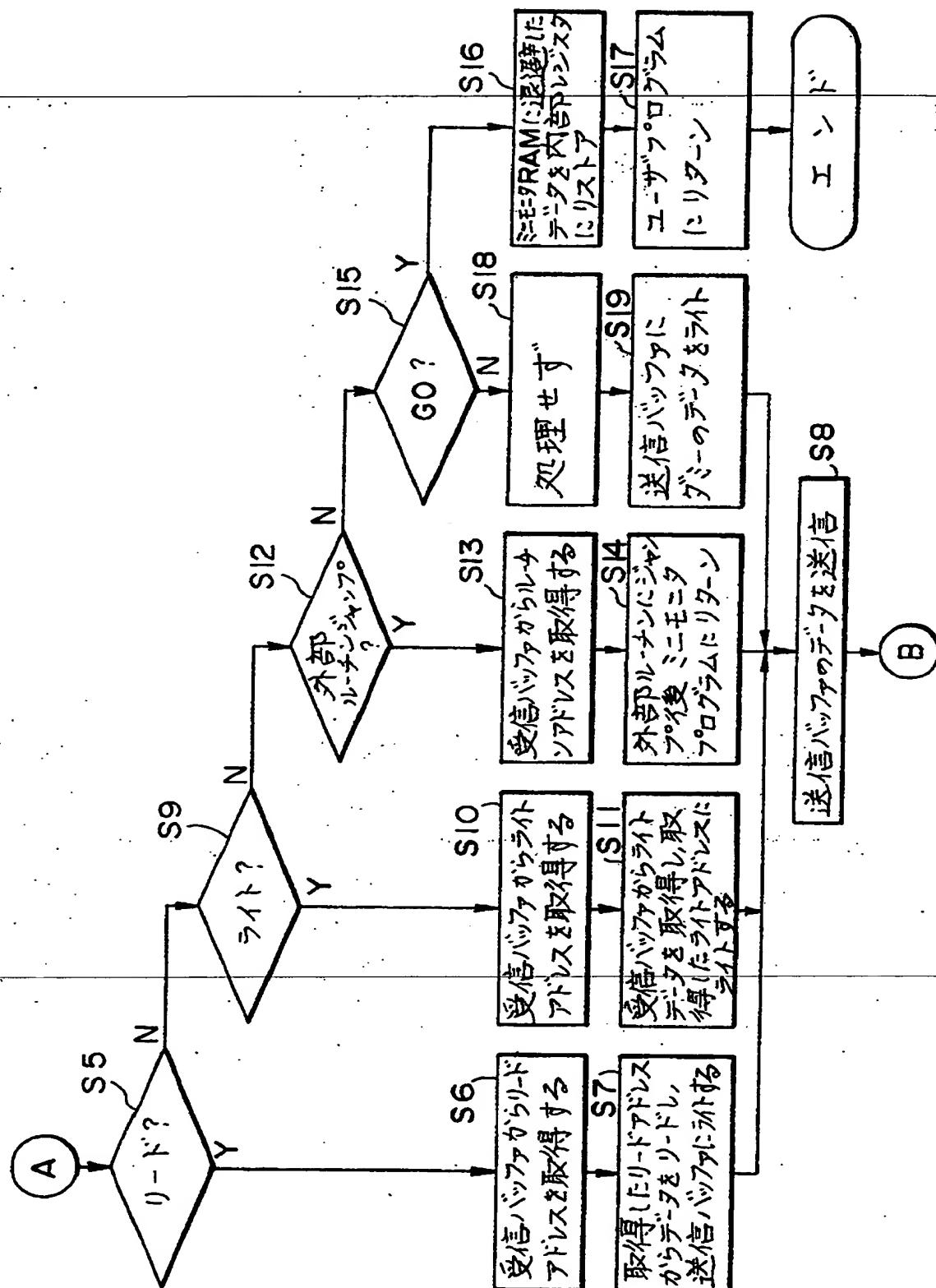
【図13】



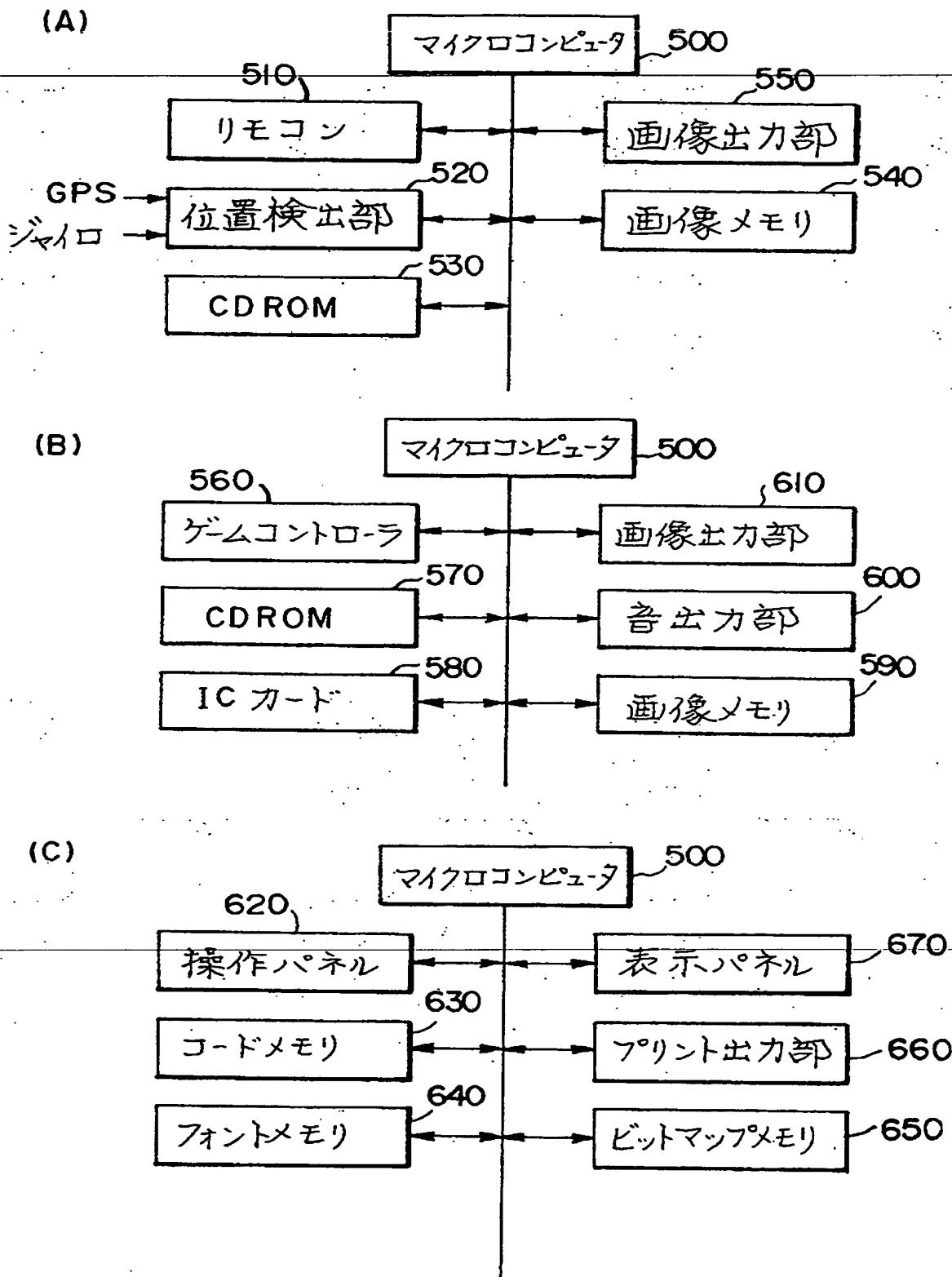
【図14】



【図15】

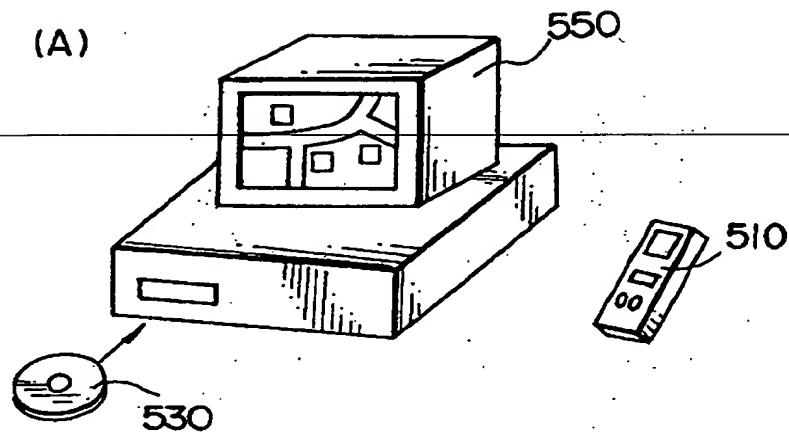


【図16】

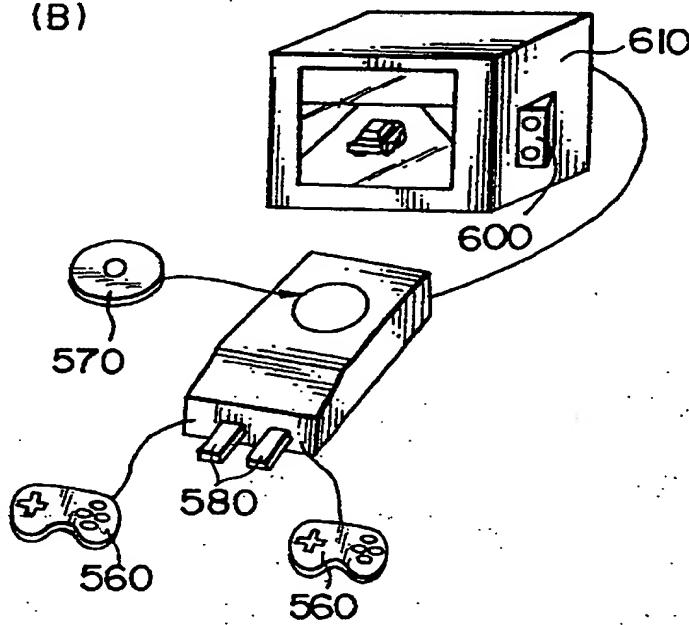


【図17】

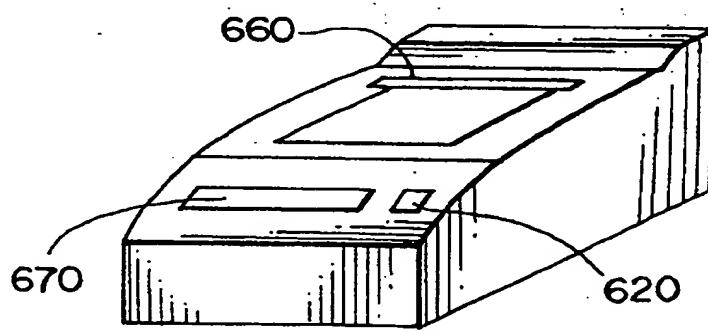
(A)



(B)



(C)



【書類名】 要約書

【要約】

【課題】 量産チップ上で強制ブレークを入力するための端子等のデバッグ時の
み必要な端子をより少なくしたマイクロコンピュータ、これを含む電子機器、及
びデバッグシステムを提供すること。

【解決手段】 ユーザーモード及びデバッグモードを有するマイクロコンピュー
タ10である。S I O D 1 6は、ユーザーモード時は強制ブレークの入力信号を
入力する端子として機能し、デバッグモード時はデバッグ情報の送受信を行うた
めの端子として機能する。また、外部のデバッグツール14が接続されていない
場合にはS I O Dをプルアップしてハイレベルに保持し、デバッグツール14に
接続することにより任意のレベル（ハイレベル又はローレベル）にできるよう
に構成する。そしてS I O D 1 6がハイレベルであるかローレベルであるかに基づ
きリセット時の実行モードを決定する。

【選択図】 図2

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000002369
【住所又は居所】 東京都新宿区西新宿2丁目4番1号
【氏名又は名称】 セイコーエプソン株式会社
【代理人】 申請人
【識別番号】 100090479
【住所又は居所】 東京都杉並区荻窪5丁目26番13号 荻窪TMビル2階 井上・布施合同特許事務所
【氏名又は名称】 井上 一
【代理人】 申請人
【識別番号】 100090387
【住所又は居所】 東京都杉並区荻窪5丁目26番13号 荻窪TMビル2階 井上・布施合同特許事務所
【氏名又は名称】 布施 行夫
【代理人】 申請人
【識別番号】 100090398
【住所又は居所】 東京都杉並区荻窪5丁目26番13号 荻窪TMビル2階 井上・布施合同特許事務所
【氏名又は名称】 大渕 美千栄

出願人履歴情報

識別番号 [000002369]

1. 変更年月日 1990年 8月20日

[変更理由] 新規登録

住 所 東京都新宿区西新宿2丁目4番1号

氏 名 セイコーエプソン株式会社